BBM201 – Data Structures – Fall 2015

Make-up Exam

28.12.2015 – 13:00

Name Surname: _____

Student ID : _____ **Section:**_____

Duration: 60 minutes

Question	1	2	3	4	5	Total
Points	24	12	14	30	20	100
Grade						

Question 1.

a) The following function calculates the number of r-combinations of n numbers, which is equal to the value of T[n,r] in the 2-dimensional array T. Find how many array accesses are made to T when the following function is executed.

```
function choose(n,r)
        for i:=0 to n-r do T[i,0]:=1;
        for i:=0 to r do T[i,i]:=1;
        for j:=1 to r do
                for i:=j+1 to n-r+j do
                        T[i,j]:=T[i-1,j-1]+T[i-1,j]
        return T[n,r]
```

b) The following code, called binsearch, searches a given number (**searchnum**) in a sorted list of **n** numbers. What is the worst-case, average-case and best-case running time of this algorithm in terms of n? Explain your answer.

```
int binsearch(int list[], int searchnum, int left, int right)
{
  int middle;
  while(left <= right){
    middle = (left + right)/2;
    switch(compare(list[middle], searchnum)){
      case -1 : left = middle + 1; break;
      case 0; return middle;
      case 1: rigt = middle –1;
    }
   }
  return -1;
}
#define COMPARE(x,y) (((x)<(y))? -1 ((x)==(y))?0:1)
```

Question 2. What is the output of the following code block if the stack consists of items {1,3,5,6,9}, where 1 is the top?

```
void enqueue(int queue[], int value);
int dequeue(int queue[]);
void push(int stack[], int value);
int pop(int stack[]);
bool isEmpty(int stack[]);



void methodA(int stack[5]){
  int queue[5],i;
  while(!isEmpty(stack))
     enqueue(queue, pop(stack));
  while(!isEmpty(queue))
     push(stack, dequeue(queue));
  while(!isEmpty(stack))
     printf("%d", pop(stack));
}
```

Question 3. Please convert the following prefix expression into infix and postfix notation:

/*-6/7+5/3+1*24-15-28

a) Infix notation:

b) Postfix notation:

Question 4. Fill the below Reverse function which reverses given singly linked list using recursion.

```
//global variables
struct Node * head;
    …

void main(){
        …
        Reverse(head);
        …
}

void Reverse (struct Node * p)
{









}
```

Question 5. According to the given array based linked list which is defined as follows:

```
#define MAX_LIST 10

typedef struct{
      char name[7];
      int link;
    }item;

item linkedlist[MAX_LIST];
int free_; //the index of the first free space in the list
int* list; //the index of the first item in the list
```

	name	link
[0]	“Banu”	1
[1]	“Irem”	4
[2]	“Zerrin”	-1
[3]	“Ali”	0
[4]	“Leyla”	6
[5]	“Ahmet"	3
[6]	“Mehmet”	2
[7]		8
[8]		9

[9]		-1

Imagine we begin with a sorted list given in the first table given below. After we insert “Adil” and delete “Leyla” in the sorted list, what will be the contents of the list? Please fill in the tables and write down the values of free_ and *list.

Initial table **Insert “Adil”** **Delete “Leyla”**

free_= 7 free_= free_=
*list = 5 *list = *list

	name	link
[0]		
[1]		
[2]		
[3]		
[4]		
[5]		
[6]		
[7]		
[8]		
[9]		

	name	link
[0]		
[1]		
[2]		
[3]		
[4]		
[5]		
[6]		
[7]		
[8]		
[9]		